



Haaga-Helia
ammattikorkeakoulu Oy

Teknologian valinta asiakkaan verkkosivuston suunnitteluun ja luontiin

Minerva Suvanto

Opinnäytetyö
Tietojenkäsittelyn
koulutusohjelma
2017



Tekijä(t) Minerva Suvanto	
Koulutusohjelma Tietojenkäsittelyn koulutusohjelma	
Raportin/Opinnäytetyön nimi Teknologian valinta asiakkaan verkkosivuston suunnitteluun ja luontiin	Sivu- ja liitesivumäärä 17 + 8
<p>Opinnäytetyötä varten sain toimeksiannon Old 7 Tattoo-yrityksen työntekijältä luoda hänelle portfoliotyylinen verkkosivusto. Ennen projektin käynnistämistä kävin läpi eri metodeita, joilla voidaan luoda tämäntyyppinen verkkosivusto.</p> <p>Vaihtoehtoisiksi sivuston luontiin valitsin sisällönhallintajärjestelmän tai luonnin alusta asti ohjelmistopinoa hyödyntäen. Sisällönhallintajärjestelmistä vertailin WordPressiä, Drupalia ja Joomlaa. Ohjelmistopinoista vaihtoehtoina oli MEAN ja LAMP.</p> <p>Päädyn valitsemaan sivustoa varten WordPressin siksi, että sillä saisi säästettyä aikaa toiminnallisten ominaisuuksien teossa ja ajan voisi käyttää ulkoasun luontiin. Opinnäytetyön lopputuloksena on sivusto, joka on julkaistu verkossa ja jonne toimeksiantaja voi ladata sisältöä.</p> <p>Tässä opinnäytetyössä kuvailen projektin vaiheet.</p>	
Asiasanat WordPress, sisällönhallintajärjestelmä, kotisivut	

Sisällys

1	Johdanto	1
1.1	Tavoitteet	1
1.2	Rajaukset	2
1.3	Käsitteet	2
2	Vaatimukset sivuston rakentamiseen	3
2.1	Käytettävyys ja käyttäjäkokemus	3
2.2	Responsiivisuus	4
2.3	Selainten väliset erot	4
2.4	Avoin lähdekoodi	5
3	Web design ja suunnittelu	6
4	Vaihtoehdot sivuston rakentamiseen	7
4.1	Sisällönhallintajärjestelmät	7
4.1.1	WordPress	8
4.1.2	Drupal	8
4.1.3	Joomla	9
4.2	Luonti ohjelmistopinolla	9
4.2.1	MEAN	9
4.2.2	LAMP	10
5	Toteutustavan valinta ja perustelut	11
6	Toteutus	13
6.1	WordPressin asennus	13
6.2	Teema ja suunnittelu	13
6.3	Domain ja SSL-sertifikaatti	15
7	Projektin tulos ja pohdinta	16
	Lähteet	18
	Liitteet	20
	Liite 1. Osa WordPress-teeman lähdekoodia	20
	Liite 2. Kuvakaappaukset sivustosta	26

1 Johdanto

Opinnäytetyön tavoitteena on rakentaa dynaaminen verkkosivusto tatuoija-yrittäjälle. Valmis lopputulos tulee käyttöön yrittäjälle omien töiden esittämiseen. Verkkosivustolla tulee olla oleellisena osana kuvagalleria, jota yrittäjä voi itse päivittää ja pitää ajan tasalla. Kuvagallerian lisäksi sivustolla tulee yhteydenottomahdollisuus, eli joko yhteystiedot näkyvillä tai yhteydenottolomake. Sivuston tarkoituksena on markkinoida yrittäjän omaa osaamista ja saada yhteydenottoja potentiaalisilta asiakkailta.

Projekti käynnistetään Old 7 Tattoo-tatuointiliikkeessä työskentelevän tatuoijayrittäjän toimeksiannosta. Yrittäjä toimii liikkeessä tällä hetkellä harjoittelijana ja on tilannut portfolio-tyylisen verkkosivuston, jonka tarkoituksena on edistää hänen tulevaa uraansa yrittäjänä. Verkkosivustolla yrittäjä voi esitellä tekemiään töitä ja saada yhteydenottoja potentiaalisilta asiakkailta. On tärkeää, että sivut ovat sellaiset, että yrittäjän on helppo päivittää itse sivustoa ja etenkin lisätä sinne kätevästi itse ottamiaan valokuvia tekemistään töistä. Verkkosivut ovat yrittäjälle nykymaailmassa välttämättömät, koska suuri osa asiakkaista etsii ja löytää tietoa verkosta ja ilman ajantasaisia verkkosivuja voi menettää asiakkaita ja liiketoiminta kärsiä. Huomioin projektin aikana asiakkaan toiveet ja ehdotukset sivuston ulkoasun suhteen, mutta muuten projektin ohjat ovat pääasiassa omissa käsissäni.

Ennen projektin käynnistämistä kartoitan eri vaihtoehtoja, joilla sivut voidaan toteuttaa. Valitsen kartoittamistani teknologioista yhden, mielestäni parhaaksi soveltuvan menetelmän, jolla toteutan projektin.

Sivuston toteuttaminen on itselleni myös ensimmäinen asiakasprojekti, jonka on tarkoitus toimia oppimiskokemuksena kohti oman yrityksen perustamista. Projekti pyritään tekemään mahdollisimman paljon asiakasta kuunnellen ja hänen kanssaan keskustellen ja yhdessä sopien. Valitsen oikean tavan toteuttaa projektin asiakkaan toiveiden ja oman harjontani perusteella.

1.1 Tavoitteet

Projektin tavoitteena on luoda ja julkaista verkkosivusto, joka tulee asiakkaan käyttöön. Opinnäytetyön tavoitteena on selvittää, millä menetelmällä aloittelevan yrittäjän kannattaa tehdä tämänkaltaisen projekti. Menetelmän valinnan jälkeen projekti toteutetaan valitulla menetelmällä ja lopuksi arvioidaan, oliko menetelmä onnistunut valinta. Valinnassa tulee ottaa huomioon ainakin projektin aikaikkuna, sivuston vaatimukset ja se, että asiakkaan tulee pystyä helposti päivittämään sivustoa itse.

Projektin ja opinnäytetyön lopputuloksena on nettisivut julkaistuna verkossa yrittäjän käytössä. Tavoitteena on luoda hyvännäköiset ja helppokäyttöiset sivut, joita on helppo lähteä kehittämään pidemmälle tulevaisuudessa.

1.2 Rajaukset

Verkkosivulle tuleva grafiikkaa ja kuvat tulevat toimeksiantajalta. Hakukoneoptimointiin tai muuhun sivuston markkinointiin liittyviin asioihin ei keskitytä tässä projektissa. Projektiin kuuluu pääsääntöisesti kaikki sellainen, mikä kuuluu sivuston suunnitteluun, toteutukseen ja julkaisuun.

1.3 Käsitteet

Alla on selitetty opinnäytetyössä käytettyjä yleisiä käsitteitä.

HTML – Hyper Text Markup Language, eli kieli, jolla tehdään HTML-websivut.

CSS – Cascading Style Sheets, eli kieli, jolla määritellään HTML-elementtien ulkonäköä.

PHP – palvelin puolen ohjelmointikieli, jolla voidaan luoda dynaamisia ja interaktiivisia sivuja.

JavaScript – web puolen ohjelmointikieli, jolla muokataan HTML-sivun rakennetta ja CSS tyylejä.

SQL-tietokanta – relaatiotietokanta, eli perinteinen tietokantamalli, joka on jäsennetty tunnistamaan relaatiot eri mallien kesken.

NoSQL-tietokanta – tietokanta, joka ei noudata relaatiomalleja, vaan esimerkiksi avain-arvo paria tai dokumenttimallia.

back-end – ohjelman tai sovelluksen taustalla toimiva logiikkataso, johon käyttäjällä ei ole suoraa pääsyä.

front-end – ohjelman tai sovelluksen käyttöliittymätaso, johon käyttäjällä on suora pääsy.

full-stack – käsite, joka pitää sisällään back-end ja front-end tasot

2 Vaatimukset sivuston rakentamiseen

Ennen varsinaisten menetelmien läpikäyntiä täytyy selvittää, mitä vaatimuksia sivustoa rakentaessa tulee huomioida. Tässä kappaleessa käydään läpi erilaisia asioita, jotka on hyvä huomioida nettisivuprojektissa. Lisäksi käsitellään hieman webkehityksen parhaita käytäntöjä. Näitä asioita tulee ottaa huomioon sivuston rakentamiseen valittavassa teknologiassa.

2.1 Käytettävyys ja käyttäjäkokemus

Käyttäjäkokemus ja käytettävyys ovat tällä hetkellä trendikäsitteitä webkehityksessä. Ne ovat todella tärkeitä asioita nettisivujen suunnitteluvaiheessa. Sivustolla on yrityksen kannalta aina jokin tavoite, esimerkiksi saada kävijä tilaamaan yrityksen uutiskirje. Verkkokaupan tavoite on taas saada asiakas ostamaan tuotteita. Tämän projektin sivuston lopullinen tavoite on tietenkin saada yhteydenottoja asiakkailta. Käytettävyydellä tarkoitetaan tässä tapauksessa sitä, kuinka hyvin käyttäjät kykenevät toteuttamaan tämän sivuston perimmäisen tavoitteen. Hyvällä käytettävyydellä luodaan parempi käyttäjäkokemus, jolla voidaan luoda yritykselle parempia asiakaskokemuksia ja tuottaa parempaa liiketoimintaa. (Brink, Gergle & Wood 2002, xii.) Käytettävyyttä määritellään muun muassa sivuston toimivuudella, helpolla opittavuudella ja muistettavuudella (Brink, Gergle & Wood 2002, 3). Nämä asiat ovat tämän projektin kannalta tärkeitä pitää mielessä suunnitteluvaiheessa.

Helppokäyttöistä sivustoa voidaan kutsua intuitiiviseksi, mikä tarkoittaa sitä, että sen käyttäjä osaa helposti navigoida sivustolla ilman ylimääräistä apua (Brink, Gergle & Wood 2002, 3). Tässä projektissa tulee huomioida sekä yrittäjän mahdolliset asiakkaat, jotka vierailivat sivustolla, että yrittäjä, joka lataa sisältöä sivustolleen. Sivuston käytön tulee olla helppoa yrittäjälle, jotta uutta sisältöä saadaan näkyville nopeasti ja näin pitää sivusto ajan tasalla. Yrittäjällä tulee siis olla käytössään yksinkertainen ja selkeä käyttöliittymä, jolla päivittää sivustoaan.

Noin vuoden mittaisen ohjelmointi urani aikana olen huomannut, että ohjelmoijan ei yleensä tarvitse miettiä kauheasti suunnitteluun liittyviä asioita. Suunnitteluun palkataan yrityksissä ulkopuolinen ammattilainen suunnittelemaan käyttöliittymiä. Tässä tapauksessa, kun tekee koko projektin itsenäisesti, täytyy ottaa huomioon myös ulkoasun suunnittelu. Varsinkin jos tekee projektin kokonaan alusta asti, täytyy admin-käyttöliittymän suunnitteluun käyttää aikaa, että se on helppokäyttöinen. Näin taataan, että yrittäjä osaa käyttää sivustoa.

Tässä projektissa ei ole sivuston kävijöiden kannalta niin paljon toiminnallisuutta, joten siihen liittyy enemmän ulkoasun suunnittelua. Tämä tarkoittaa sitä, että asiat ovat näkyvillä loogisesti, ja kävijät eivät hämmenny ja löytävät tarvittavat tiedot ilman suurempaa vaivaa.

2.2 Responsiivisuus

Jos sivuston luonnin aikana ei ole mietitty ollenkaan eri näyttöjen resoluutioita, tai yksinkertaisemmin laitteen kokoa, sivusto saattaa näyttää hyvältä omalla laitteella mutta näyttää todella erilaiselta eri käyttäjän laitteella. Tähän vaikuttaa osittain eri selainten erot, joista kerron myöhemmässä kappaleessa, ja käyttäjän näyttöasetukset, mutta suurimaksi osaksi tähän vaikuttaa laite itsessään. Näitä asioita on muun muassa resoluutio ja värientoisto mahdollisuudet. (Robbins 2006, 27.) Sivuston tulee olla yhteensopiva niin pöytäkoneella kuin mobiililaitteillakin.

Mobiilikäyttö tulee huomioida siksi, että monet omistavat nykyään älypuhelimien, jolla he vierailevat sivustolla heti siitä kuultuaan. Responsiivisuus tulee huomioida jo sivuston suunnitteluvaiheessa (*mobile-first-design*) jos haluaa nykyaikaiset sivut. Mobiililaitteet ovat olleet jo pitkään nopeasti kasvava osa tietotekniikan ja päätelaitteiden myynnissä ja älypuhelimien myynti ylittää jo tietokoneidenkin myynnin. Koska laitteita on erilaisia ja yhä enemmän, web suunnittelijoiden on hyvin vaikea ennustaa sivuston vierailijoiden käyttämää laitetta. Tämän takia täytyy siis pystyä toteuttamaan sellainen sivusto, joka toimii mahdollisimman monenlaisilla eri resoluutioilla. (Gardner 2011, 14.)

Responsiiviseen suunnitteluun kuuluu kolme keskeistä osaa.

- ulkoasu, joka skaalautuu näytön mukaan (*fluid layout*)
- kuvat, jotka toimivat skaalautuvassa kontekstissa
- kyselyt (*media query*), joita käytetään optimoimaan ulkoasua eri näytöille.

(Gardner 2011, 14.)

Valitun teknologian tulee olla sellainen, jolla on mahdollisuus tehdä sivuista toimivat myös mobiilissa.

2.3 Selainten väliset erot

Laitteen resoluution lisäksi sivuston ulkoasuun vaikuttaa paljon eri verkkoselaimet. Vaikka sivusto näyttäisi hyvältä omalla lempiselaimella, se voi näyttää rikkiäiseltä jollain muulla selaimella. Selaimen taustalla toimii selainmoottori, joka on koodi, joka kertoo selaimelle, kuinka näyttää ja muuntaa sisältö ja saatavilla olevat tyylitiedot näytölle. Moottori hoitaa sen, miten esimerkiksi CSS-tyylitiedostoissa määritellyt tyylit näytävät sivustolla. (Robbins 2006, 14.)

Webkehittäjien tulee tietää tämän hetken suosituimmista selaimista, eikä ainoastaan omista suosikeistaan (Robbins 2006, 14). Esimerkiksi statcounter.com-sivustolta löytyy ajankohtaista tietoa eri selainten suosiosta. Oman sivuston kävijöiden suosimia selaimia voi myös seurata analytiikan avulla, ja näin optimoida sivuja aina suosituimpien selainten mukaan. Nyrkkisääntönä on kuitenkin tarjota hyvät nettisivut toimimaan kaikilla suosituimmilla selaimilla (Robbins 2006, 14).

Chrome	IE	Firefox	Safari	Opera
54.89%	3.69%	5.9%	14.88%	1.2%

08/2017 Webkäyttäjien suosimat selaimet, StatCounter

Sivut tulee optimoida siis toimimaan ainakin edellä mainituilla suosituimmilla selaimilla. Tätä vaihetta tehdessä on tärkeää testata eri selaimilla, miltä sivut näyttävät (Robbins 2006, 25). Optimointia tärkeämpää on kuitenkin pitäytyä webstandardeissa, koska suosittu selaimet tukevat näitä standardeja. Tämä tekee selainten välisten erojen käsittelystä paljon yksinkertaisempaa. (Robbins 2006, 22.)

2.4 Avoin lähdekoodi

Avoin lähdekoodi mahdollistaa kehityksen, joka on läpinäkyvää ja muiden kehittäjien arvioitavana. Avoimella lähdekoodilla tavoitellaan parempilaatuista, luotettavampaa, joustavampaa koodia halvemmalla hinnalla. (Opensource, 2017.)

Ohjelmaa, joka on avointa lähdekoodia, tulee voida käyttää mihin tahansa tarkoitukseen. Kehittäjällä tulee olla mahdollisuus ja vapaus opiskella lähdekoodia ja muuttaa sitä toimimaan haluamallaan tavalla. Avointa lähdekoodia saa jakaa vapaasti muille, alkuperäisenä sekä muokattuna. (GNU. 2017.)

Tässä projektissa valitun teknologian tulee olla lähtökohtaisesti vapaata lähdekoodia. Tähän on syynä se, että projektista ei kertyisi ylimääräisiä kustannuksia eikä huolta tekijänoikeuksien rikkomisesta. Projekti pyritään tekemään mahdollisimman pienillä kustannuksilla.

3 Web design ja suunnittelu

Olen useasti todennut koulussa, töissä ja yleisesti elämässä, että hyvännäköinen design on oleellinen osa mitä tahansa tuotetta. Esimerkiksi kun olin vaihdossa Tanskassa, tiesin heti lähteä suunnittelemaan saamaamme toimeksiantoa ulkonäön kannalta. Ryhmämme saikin lopulta toimeksiantajalta hyvää palautetta selkeästä ja hyvännäköisestä käyttöliittymästä. Toki tuotteen tulee olla toimiva mutta hyvällä designilla voi saada joitain pienempiä heikkouksia anteeksi.

Websivujen suunnitteluun vaikuttaa aiemmissa kappaleissa kuvatut asiat, kuten käyttökokemus ja eri selaimet sekä päätelaitteet. Lisäksi suunnittelussa on hyvä ymmärtää sivuston kävijöiden toiminnasta ja käyttötottumuksista. Länsimaisessa kulttuurissa luetaan asiat vasemmalta oikealle ja ylhäältä alas. Tämän takia tärkeät asiat tulisi sijoittaa ylhäälle ja vasemmalle, jotta ne ovat ensimmäiset asiat, jotka kävijä näkee. (Miller, 2011.)

Jokainen sivu näyttää toki aina erilaiselta värien ja typografian kannalta, mutta käyttäjien toiminta on vaikuttanut websuunnitteluun siten, että on kehittynyt tiettyjä rakenteellisia käytäntöjä. Websivuston osia ovat esimerkiksi ylätunniste (*header*), navigaatio, sisältö ja sivupalkit sekä alatunniste (*footer*). Nämä elementit on usein aseteltu siten, että ne ovat kävijälle luonnollisessa järjestyksessä. (Miller, 2011.)

Ylätunniste esittelee sivuston brändin ja näkyy jokaisella sivulla. Tämä osio on sivustolla ensimmäisenä, joten se sisältää kaikki tärkeimmät tiedot, kuten sivuston nimen ja logon. Sivuston sisältö on oleellisin osa sivua ja se osa, missä kävijän tulisi viettää suurin osa aikansa. Jos sisältö on erityisen pitkä, se olisi hyvä jakaa esimerkiksi otsikoilla tai tyhjällä tilalla. (Miller, 2011.)

Perusrakenteen lisäksi ulkoasun tärkeitä asioita ovat värimaailma ja typografia. Värimaailma ja fontti tullaan sopimaan täysin asiakkaan mieltymysten mukaan. Typografiassa on hyvä muistaa, että fonttikoko on tarpeeksi suuri ja kappaleiden välillä on tyhjää tilaa. Näin sivustolla näkyvää tekstiä on helppo lukea.

4 Vaihtoehdot sivuston rakentamiseen

Nettisivujen tekemistä varten on paljon eri tapoja perinteisen HTML ja CSS yhdistelmän lisäksi. On tarjolla erilaisia menetelmiä ja viitekehyksiä, joilla on tietenkin eri tavoitteet. Tässä kappaleessa käydään läpi mitä eri tapoja voidaan valita nettisivuprojektin tekemiseen. Tavoitteena on valita tekniikoista yksi, mitä käytetään projektissa, että se on paras tavoitteiden saavuttamiseksi.

Projektin suunnitteluun ja toteutukseen on varattu aikaa neljä kuukautta, jolloin projekti toteutetaan täysipäiväisen työn ohella, eli iltaisin ja viikonloppuisin. Valitun teknologian tulee siis tukea myös aikataulua.

Keskeisiä kysymyksiä valintaprosessissa ovat:

- Saadaanko tällä tekniikalla projekti valmiiksi aikataulun puitteissa?
- Meneekö turhaa aikaa uuden teknologian opettelemiseen ja tuoko uuden teknologian käyttäminen lisäarvoa projektille vai onnistuuko sama jollain yksinkertaisemmalla tavalla?

Teknologian valinnasta vastaa tekijä, koska asiakas ei osaa tässä tapauksessa ottaa kantaa tekniseen puoleen. Teknologian tulee kuitenkin olla sellainen, millä saadaan asiakkaan toiveiden mukaiset sivut. Koska sivut kuitenkin edustavat yrittäjää, täytyy niiden myös olla hänen oman näköisensä. Ulkoasuksi ei valita valmista ulkoasua, joita on tarjolla esimerkiksi sisällönhallintajärjestelmissä.

Valmis sivusto julkaistaan Linux-palvelimelle, joten teknologian tulee olla sellainen, joka toimii sillä alustalla.

4.1 Sisällönhallintajärjestelmät

Sisällönhallintajärjestelmä on ohjelma, joka seuraa sivuston koko sisältöä ja pitää sen hallittavissa. Sisältöä ovat esimerkiksi tekstit, kuvat, musiikki, videot, tiedostot ja kaikki muu, mitä sivustolla on. Sisällönhallintajärjestelmän käytön suurin hyöty on se, ettei sen käyttäjä tarvitse teknistä osaamista tai tietoa tarvittaessa oikeastaan ollenkaan. (Joomla 2017.)

Sisällönhallintajärjestelmää voi käyttää myös esimerkiksi verkkokaupan ylläpitämiseen, joten eri käyttötarkoituksiin soveltuvia järjestelmiä löytyy paljon. Tässä kappaleessa on kuvattu kolme suosituinta sisällönhallintajärjestelmää, jotka soveltuvat tämän projektin kaltaisten nettisivujen tekemiseen: WordPress, Drupal ja Joomla. Sisällönhallintajärjestelmän

suurin etu on se, että asiakas pääsee helposti muokkaamaan verkkosivustonsa sisältöä, ja käyttöliittymä on tätä varten valmis ja toimiva.

4.1.1 WordPress

WordPress on luotu alun perin blogialustaksi, mutta on kehittynyt siitä suuremmaksi sisällönhallintajärjestelmäksi. WordPressiin on saatavilla runsaasti eri plugineita, widgettejä ja teemoja, joita löytyy niin ilmaisia kuin maksullisiakin. Valmiiden komponenttien lisäksi WordPressiin voi tehdä kaikki nämä asiat itsekin. (WordPress 2017.)

WordPressillä on myös palvelu wordpress.com, jossa voi luoda blogin nopeammin. Tässä palvelussa ei kuitenkaan ole yhtä laajaa muokausmahdollisuudet kuin WordPressissä, mutta se on hyvä vaihtoehto, jota suosittelisin, jos asiakas haluaisi esimerkiksi tehdä sivuston itse.

BuiltWith-sivustolta (2017), jossa kerrotaan millä alustoilla verkkosivuja on rakennettu, nähdään, että WordPress on käytetyin sisällönhallintajärjestelmä. WordPressin omilla sivuilla (2017) kerrotaan, että WordPressiä käytetään ylläpitämään 28 prosenttia netin tarjonnasta. Suosiota tälle alustalle siis löytyy.

WordPressin teeman rakentamiseen tarvitaan PHP, CSS ja HTML osaamista. WordPressin suosio sisällönhallintajärjestelmänä tuo myös kehittäjälle etuna sen, että siitä löytyy paljon ohjeita ja tutoriaaleja netistä. (WordPress 2017.) Tämä on etu kehittäjälle, kun on hyvät resurssit ja apukeinot kehittää sivustoja.

4.1.2 Drupal

Drupal on WordPressin tapaan sisällönhallintajärjestelmä, jota voidaan käyttää monen erilaisen sovelluksen luomiseen. Drupalissa on helppo hallita sivuston sisältöä, siinä on luotettava suorituskky ja hyvä tietoturva ja sen modulaarisuus tekee siitä joustavan alustan. (Drupal 2017.) WordPressin tapaan Drupalille löytyy paljon eri plugineita ja teemoja, joita voi itse muokata haluamakseen. Drupal on kolmanneksi suosituin sisällönhallintajärjestelmä. (Mening 2015.)

Drupalia käytetään monissa suurissa sivustoissa, esimerkiksi USAn White House:n ja BBC:n sivut on luotu Drupalilla. (Drupal 2017.) Drupalilla onkin hyvä tehdä suuremmat sivut, koska sillä luodut sivut latautuvat nopeammin kuin WordPressillä tai Joomlailla luodut sivut. Tämä johtuu siitä, että Drupal käyttää vähemmän palvelimen resursseja. Drupalin käyttö vaatii hyvän osaamisen HTML, CSS ja PHP-ohjelmoinnista. (Mening 2015.)

4.1.3 Joomla

Joomla on toiseksi suosituin sisällönhallintajärjestelmä. Se on tarpeeksi tehokas ajaakseen useimpia sivuja ongelmitta ja siihen ei kuitenkaan tarvitse yhtä paljon teknistä osaamista kuin Drupalilla. Kuten aiemmassa kahdessa järjestelmässä, Joomlaan saa plugineita ja teemoja ja yleisesti ominaisuudet ovat hyvin samankaltaisia. (Mening 2015.)

Joomla on erityisen hyvä esimerkiksi verkkokaupan rakentamiseen. Tämä onnistuu myös WordPressillä ja Drupalilla, mutta Joomlaan ei tarvitse nähdä yhtä paljon vaivaa ja ei tarvitse räätälöidä niin paljoa. (Mening 2015.)

4.2 Luonti ohjelmistopinolla

Sisällönhallintajärjestelmän lisäksi vaihtoehtona on rakentaa sivusto kokonaan itse. Tähän avuksi ja kehityksen nopeuttamiseksi on hyvä valita jokin ohjelmistopino (*software stack*). Ohjelmistopinolla tarkoitetaan eri sovelluksia, jotka toimivat kohti yhtenäistä tulosta. Tässä tapauksessa pinoon kuuluu neljä osaa: tietokanta, back-end-kehys, front-end-kehys ja käyttöjärjestelmä. Ohjelmistopinon avulla voidaan kehittää nopeasti ja säästää aikaa ja nopeuttaa kehitysprosessia. Ohjelmistopinoissa kaikki komponentit toimivat keskenään oikein, eikä näin ollen tarvitse itse valita ja miettiä, mitkä toimivat keskenään hyvin. (Wodehouse 2017.)

Asioita, jotka tulee ottaa huomioon ohjelmistopinon valinnassa, on esimerkiksi tuotteen mahdollinen jatkokehitys ja se, miten paljon se mahdollisesti tulee kasvamaan ajan saatossa. Teknologian tarjoama latausnopeus on hyvä pitää mielessä juurikin sivuston koon huomioiden. Lisäksi tulee miettiä, millainen tietokanta on paras omalle projektille. Viimeisenä mielestäni tärkeänä asiana kannattaa myös miettiä omaa osaamistaan ja sitä, kuinka suuri oppimiskäyrä jossain mahdollisesti uudessa teknologiassa on ja kuinka paljon aikaa siihen uhrautuu. (Wodehouse 2017.)

Tässä kappaleessa on kerrottu kahdesta eri ohjelmistopinosta, joilla sivuston toteutus olisi mahdollista.

4.2.1 MEAN

MEAN on nykyaikainen ohjelmistopino, joka toimii täysin JavaScript-kielellä. Tämä on pinnon suurin hyöty itselleni, koska minulla on jo aikaisempaa kokemusta JavaScriptistä. Pinnon kuuluu MongoDB-tietokanta, joka on joustava JSON:illa toimiva NoSQL-tietokanta.

Tietokannan lisäksi pinoon kuuluu AngularJS front-end framework, jolla voidaan tehdä näyttäviä websivuja, Express.js, joka on web framework, joka pyörii Node.js alustalla. Node.js toimii tässä tapauksessa palvelimen roolissa. (Wodehouse 2017.)

MEAN-pinon hyötyjä ovat Node.js-alustan mukana tuomat JavaScript moduulit ja kirjastot sekä paketit. AngularJS-frameworkin joustavuus takaa mobiiliin skaalautuvien sivujen kehityksen. Suurin etu on kuitenkin siinä, että kaikissa tasoissa käytetään samaa kieltä, eli JavaScriptiä. NoSQL tietokanta myös säästää aikaa tietokannan rakentamisessa. MEAN on myös täysin avointa lähdekoodia. (Wodehouse 2017.)

MEAN-pinossa eniten itseäni kiinnostaa sen AngularJS-framework, jota olen käyttänyt palkkatyössäni ja todennut sen tehokkaaksi front-end työkaluksi.

4.2.2 LAMP

LAMP on avointa lähdekoodia oleva webkehityksen alusta. LAMP on lyhenne alustan eri tekniikoista. Linux toimii tässä käyttöjärjestelmänä, Apache web-palvelimena, MySQL tietokantana ja PHP alustan ohjelmointikielenä. PHP-kielen sijaan joskus käytetään myöskin Perl- tai Python-kieltä. (Wodehouse 2017.) Jos tämä teknologia valittaisiin projektin toteuttamiseen, ohjelmointi tapahtuisi kuitenkin PHP:lla, koska sen opettelemiseen olen näistä kolmesta kielestä eniten tutustunut.

Tällä pinolla on mahdollista luoda helposti erityisesti dynaamisia websivuja ja sovelluksia. LAMP on yksi perinteisimmistä ohjelmistopinoista ja näin ollen erittäin vakaa valinta. LAMP-pinon hyötyjä tätä projektia ajatellen on helppo kehitys ja käyttöönotto, tietoturvallisuus ja avoimen teknologian tuomana hyötynä suuri kehitysyhteisö, joka takaa helpon avun. Pinossa käytettävä SQL-tietokannan avulla voidaan myös järjestellä suuria määriä monimutkaista dataa. (Wodehouse 2017.) Tämä ei kuitenkaan ole välttämättä oleellista kyseisessä projektissa, koska siihen ei kuulu monimutkaisia datarakenteita.

LAMP-pinon otin mukaan mahdolliseksi teknologiaksi siksi, että se tosiaan on perinteisin vaihtoehto tehdä dynaamiset verkkosivut Linux -alustalle.

5 Toteutustavan valinta ja perustelut

Toteutustavan valinnassa tulee ottaa huomioon ainakin sivuston vaatimukset, sivuston mahdollinen kehitys tulevaisuudessa ja tekijän tekninen osaaminen. Alla olevassa taulukossa on tiivistetty, mitkä aiemmissa kappaleissa kuvatut teknologiat sisältävät projektin kannalta tärkeimmät vaatimukset.

Teknologia	Valmis käyttöliittymä	Tietoturva	Linux	Avoin lähdekoodi
WordPress	x	x	x	x
Drupal	x	x	x	x
Joomla	x	x	x	x
MEAN			x	x
LAMP		x	x	x

Taulukosta nähdään, että kaikki teknologiat toteuttavat avoimen lähdekoodin vaatimuksen sekä Linux-palvelimella toimivuuden. Tietoturva sisältyy suoraan kaikkiin muihin teknologioihin paitsi MEAN-pinoon. Sisällönhallintajärjestelmät WordPress, Drupal ja Joomla ovat ainoat teknologiat, jotka sisältävät valmiin admin-käyttöliittymän.

Kaikki tekniikat soveltuisivat näillä perusteilla hyvin projektin toteuttamiseen. Ohjelmistopion ja sisällönhallintajärjestelmän välillä päädyin kuitenkin valitsemaan jälkimmäisen.

Alusta asti itse rakentaminen olisi oman oppimisen kannalta hyödyllistä, ja ajattelinkin sen olevan valintani ennen eri menetelmien selvittämistä. En kuitenkaan kokenut sen tuovan asiakkaalle minkäänlaista lisäarvoa. Lisäksi valmis käyttöliittymä ja back-end toiminnot, jotka sisällönhallintajärjestelmä tarjoaa, sopivat paremmin projektin aikatauluun. Näin voin keskittyä paremmin itse sivuston ulkoasun toteuttamiseen. Sisällönhallintajärjestelmällä ei tarvitse myöskään erikseen keskittyä liikaa toteuttamaan tietoturvallista sivustoa, koska nämä ominaisuudet ovat valmiina.

Kolmesta sisällönhallintajärjestelmästä päädyin valitsemaan WordPressin projektin toteutukseen. Itselläni on hieman aikaisempaa kokemusta WordPress-alustasta blogin luontiin ja hieman oman teeman tekemisestä, joten WordPressin ominaisuudet ja käyttömahdollisuudet ovat minulle jokseenkin tuttuja. Tämän takia koin, että sen syvempi opettelu säästää aikaa sivujen luonnissa. WordPressin avulla on helppoa jatkokehittää sivustoa tulevaisuudessa ja lisätä siihen uusia ominaisuuksia. WordPressin suurin haaste tulee olemaan

PHP-ohjelmoinnin opettelu, sillä siitä ei ole hirveästi kokemusta, eli joka tapauksessa jostain uutta tästä projektista tulen oppimaan. Uskon, että tällä valinnalla säästän kuitenkin huomattavasti aikaa samanlaisen lopputuloksen saavuttamiseksi.

Asiakkaalla oli muutama toive ennen projektin aloitusta. Sivut tehdään englanniksi. Pääosassa ovat tietenkin kuvat. Ulkoasun suhteen sain aluksi vapaat kädet suunnitteluun ja projektin edetessä käymme yhdessä läpi ulkoasuun liittyviä seikkoja. Asiakas toimittaa kuvat ja logot jotka hän haluaa sivustolleen näkyviin. Yhteistyöstä asiakkaan kanssa olen kertonut enemmän seuraavassa kappaleessa.

6 Toteutus

Tässä kappaleessa on kuvattu projektin toteutuksen vaiheet ja hieman suunnittelusta asiakkaan kanssa. Projektia alettiin tekemään toimeksiantajan alkuperäisten toiveiden mukaan pian sopimuksen allekirjoittamisen jälkeen.

6.1 WordPressin asennus

WordPressin asennuksen tein suoraan harrastuspalvelimelle WordPressin omilta sivuilta löytyvien ohjeiden mukaan. Asennuksen olen tehnyt kerran aikaisemmin, mutta ohjeet toimivat hyvänä muistutuksena, miten asiat tulee tehdä. Tietokannan luontia varten ei ollut itselläni oikeuksia palvelimella, joten palvelimen omistaja loi tietokannan ja käyttäjän tietokantaa varten.

WordPressin asennuksen vaiheet:

1. WordPress paketin lataaminen ja purkaminen paikallisesti
2. Tietokannan luonti
3. wp-config-sample.php-konfigurointitiedoston nimeäminen wp-config.php:ksi ja tietokanta-tietojen lisääminen tiedostoon (kannan nimi, käyttäjä, salasana ja palvelin, jolla kanta on)
4. WordPress tiedostojen siirtäminen palvelimelle luotuun kansioon
5. WordPressin asennusskriptin ajaminen navigoimalla url-osoitteeseen, johon tiedostot siirrettiin

(WordPress 2017.)

Asennus ei ollut monimutkainen eikä siihen mennyt kauheasti aikaa. Asennuksen jälkeen valmistelin sivustoa hieman admin-käyttöliittymästä, eli loin tarvittavat sivut, poistin esimerkkinä toimineet blogipostaukset ja widgetit sekä valitsin etusivulla näytettävän sivun.

6.2 Teema ja suunnittelu

Projektia varten tein itse WordPress-teeman. Aloitin teeman suunnittelun piirtämällä ensiksi rautalankamallin paperille. Suunnitelma oli aika yksinkertainen ja suunnittelu-kappaleessa mainitun ulkoasun mukainen siten, että elementit kulkevat oikeassa järjestyksessä. Etsin inspiraatiota myös muun muassa muiden tatuointisivustojen ulkoasuista, jotta saisin käsitystä siitä, millaisia sivuja he tarjoavat erityisesti kuvagallerian suhteen. Kuvagalleriassa oleellista oli mielestäni se, että esikatselukuvat näkyvät tarpeeksi suurena, ja halutessaan kuvan voi sitten avata vielä suuremmaksi. Rautalankamallin jälkeen toteutin kärkeän version teemasta, jonka näytin asiakkaalle. Asiakkaan kanssa keskustelimme yhdessä parannusehdotuksista, värimaailmasta ja sivustolla käytettävästä fontista. Tämän pohjalta lähdin muokkaamaan sivustoa oikeaan suuntaan.

Ensimmäiseen versioon olin tehnyt etusivun, jossa oli tarkoituksena esitellä yrittäjä, galleria-sivu ja yhteydenotto-sivu. Ensimmäisen tapaamisen jälkeen asiakas toivoi, että etusivuna toimisi suoraan kuvagalleria, jotta oleellinen sisältö olisi heti näkyvillä sivustolle saapuessa. Tapaamisen jälkeen päätin lisätä myös sivulle alatunnisteen, jossa olisi näkyvillä yhteystiedot. Näin yhteystiedot olisivat näkyvillä koko ajan. Ensimmäiseen versioon olin myös lisännyt WordPress galleria-pluginin. Asiakas kuitenkin pyysi, jos gallerian voisi näyttää suoraan hänen Instagram-feedistään. Ensimmäisen tapaamisen pohjalta päädyin poistamaan siis erillisen etusivun ja vaihtamaan gallerian Instagram-feediin.

Värimaailman kannalta asiakas toivoi mustavalkoista, ja olin itse samaa mieltä. Mustavalkoinen ulkoasu ei veisi huomiota pois kuvagalleriasta, ja jos sivusto olisi itsessään liian värikäs, kuvat eivät olisi niin hyvin esillä sivuilla.

Toisessa tapaamisessa kävimme lähinnä läpi teknisiä seikkoja, kuten domainin ostamisen ja asiakkaan Instagram-tilin kuvien lisäämisen sivustolle. Tätä varten tarvitsin *access tokenin*, eli oikeuden käyttää asiakkaan tilillä olevia kuvia. Tähän tapaamiseen olin myös tehnyt aiemmassa tapaamisessa sopimamme muutokset, ja asiakas kertoi sivuston näyttävän jo paljon paremmalta. Samalla hän myös toimitti sivuja varten oman logonsa.

Kolmannessa ja viimeisessä tapaamisessamme kävimme läpi vielä muutamia pieniä asioita. Asiakas toivoi mahdollisimman vähän vieritettävää sivustoa, ja pyysi, voiko kuvien näytettävää määrää tästä syystä vähentää, koska sivulla oli kuitenkin ”näytä lisää kuvia”-toiminto. Tähän versioon olin lisännyt myös yhteydenottolomakkeen. Lomakkeen ulkoasussa ei ollut muuta ongelmaa kuin kenttien kokojen pienentäminen.

Asiakkaan kanssa suunnitellessa sai paljon hyviä, uusia näkökulmia. Kun tekee sivustoa, tuntuu välillä, että tietyille asioille tulee niin sanotusti sokeaksi, joten on hyvä saada palautetta suoraan asiakkaalta. Tapaamisten lisäksi olin yhteydessä asiakkaaseen tekstiviestitse, jos kysyttävää tuli.

Suurin osa projektiin käytetystä ajasta meni teeman luomiseen. Teeman luonnissa pääsin alkuun ohjeilla, joissa oli valmiit pohjat teematiedostoista. (Dachis 2012.) Apua löytyi kehityksen aikana helposti esimerkiksi stackoverflow-foorumilta.

6.3 Domain ja SSL-sertifikaatti

Avustin asiakasta ostamaan itselleen domainnimen, jotta hän saa itselleen haluamansa url-osoitteen. Asetin domainin ohjautumaan oikeaan kansioon palvelimella tekemällä virtual host-tiedoston. Virtual host ohjaa domainin oikeaan paikkaan. Lisäsin sivustolle vielä SSL-sertifikaatin, eli suojatun yhteyden varmenteen. Sertifikaattia varten ajoin valmiiksi tehdyn skriptin ajan säästämiseksi. Nämä ovat asioita, jotka hoituivat helposti, koska olen tehnyt niitä jo ennen tätä projektia.

7 Projektin tulos ja pohdinta

Teeman luominen oli helpompaa, kuin aluksi ajattelin. PHP-osaaminen ei ollut niin olennaista, ja sitä tuli kirjoitettua teeman lähdekoodiin lopulta vain muutama rivi. Sivuston ylläpitäminen on helppoa ja uusia ominaisuuksia voi tarvittaessa lisätä nopeasti, ja muokata niitä mieleisikseen teema-tiedostojen kautta.

Asiakas toivoi projektin puolesta välissä, että hänen ei itse tarvitse koskea WordPressin admin-käyttöliittymään. Tähän ratkaisu oli luoda Instagram-feed erillisen WordPress kuvagallerian sijaan, jotta hän voi päivittää kuvat Instagramin kautta. Asiakkaan mahdollisuutta päästä itse muokkaamaan sivuja WordPressin kautta ei kuitenkaan ole rajattu pois. Näin asiakas voi käydä itse päivittämässä sivuston sisältöä halutessaan. Sovimme kuitenkin, että aluksi voin tehdä pienet muutokset, kuten tekstien päivittämisen itse. Näin jälkepäin ajateltuna, gallerian ja kuvien lataamisen vaihduttua Instagram-feediksi, itse tehty sivusto olisi voinut olla aivan yhtä hyvä vaihtoehto.

Teknologian valinta oli kuitenkin ajallisesti sopiva aikataulussa pysymisen kannalta. Tein projektin täysipäiväisen työn ohella, ja alusta asti rakennettu sivusto olisi vienyt huomattavan paljon enemmän aikaa. Aikaa olisi tuhlautunut ympäristön asentamiseen ja uusien tekniikoiden opetteluun, eikä projekti olisi todennäköisesti valmistunut määräaikaan mennessä.

Suurin WordPressin ja itse tehdyn sivuston ero oli kokemukseni mukaan ympäristön asentaminen. WordPressissä tämä oli erittäin yksinkertaista, ja antoi mahdollisuuden päästä heti rakentamaan ulkoasua ja muuta oleellista. Yksinkertaisen WordPress-teeman tekeminen toi tietenkin rajoitteita siihen, miten ulkoasu muotoutuu, koska teemassa on valmiit kohdat tietyille osille. Siitä ei tosin tässä projektissa ollut erityisemmin haittaa. Aikaa säästy, kun ei tarvinnut itse tehdä back-end toimintoja. Tekemistä kuitenkin riitti ulkoasun parissa.

Sivuston suunnittelu oli siinä mielessä vaikeaa, että olen töissä tottunut tekemään ulkoasuja annetun suunnitelman mukaan, sen sijaan, että saisin itse päättää ulkoasusta. Vapaus ja vastuu sivuston ulkonäöstä loivat paineita. Tämä oli kuitenkin hyvä tapa päästä käyttämään omaa mielikuvitusta ja harjoitella myös suunnitteluprosessia. Alusta asti oli kuitenkin selvää, että halusin sivuista asiakkaan näköiset, joten siksi pyrin ottamaan asiakasta mukaan suunnitteluprosessiin mahdollisimman paljon.

Ajatuksenani on ollut aloittaa oma yritys tai freelancer-toiminta nettisivuja tehden, ja tämän projektin tarkoitus oli saada ensituntumaa asiakasprojekteista. Jälkeenpäin voin todeta, että sivutyönä tai harrastuksena tämä ei tule toimimaan ainakaan lyhyillä aikatauluilla, sillä projekti vaati keskittymistä ja aikaa.

Ennen opinnäytetyön tekemistä olin kallistunut tekemään sivuston itse alusta asti.

WordPressin käyttöönotto oli kuitenkin positiivinen kokemus, ja voin kuvitella valitsevani sen muihin samankaltaisiin projekteihin tulevaisuudessa. Ohjelmistopinoa hyödyntäen sivun tekeminen vaatisi todennäköisesti valmiin pohjan tekemistä omaan käyttöön, joka vie paljon aikaa. Oman yrityksen pystyttämisympäristössä tämä tulee kuitenkin varmasti tapahtumaan, ja opinnäytetyö olisi toisaalta ollut siihen hyvä ponnahduslauta. Ajallisesti se ei olisi kuitenkaan ollut realistinen tavoite.

Projektin aikataulu oli mielestäni realistinen, siihen nähden, että tein täysipäiväisesti töitä muualla. Itsenäinen projektityöskentely vaati paljon omistautumista projektille. Jos päättäisin tehdä samanlaisia projekteja tulevaisuudessa, sen tulisi olla täysipäiväistä työtä. Eniten projektin aikana jäi harmittamaan se, etten pystynyt käyttämään aivan niin paljon aikaa sen tekemiseen, kuin olisin halunnut. Muutamat pienet asiat jäivät häiritsemään minua sivulla, kuten se, että Instagram-pluginissa on mahdollista avata kuvat vain uudessa välilehdessä, eikä suoraan sivuilla. Esimerkiksi tästä keskustelimme asiakkaan kanssa, mutta päädyimme siihen lopputulokseen, että pluginin voi tehdä myöhemmin itse toimimaan haluamallaan tavalla. Tärkeintä on kuitenkin se, että asiakas on tyytyväinen sivustoonsa ja jatkokehitysmahdollisuus on tehty helpoksi.

Lähteet

BuiltWith. 2017. Luettavissa: builtwith.com. Luettu: 15.7.2017.

Brink, T., Gergle D. & Wood S. 2002. Usability for the Web: Designing Web Sites that Work. Academic Press. Lontoo.

Dachis A. 2012. How to Create a Custom Theme for Your WordPress Blog with Minimal Coding Required. Luettavissa: <https://lifehacker.com/5890139/how-to-create-a-custom-theme-for-your-wordpress-blog-with-minimal-coding-required>. Luettu: 15.7.2017.

Gardner, B. 2011. Responsive Web Design: Enriching the User Experience. Sigma (Noblis Technical Journal), Volume 11 Number 1, s.13-19.

GNU. 2017. What is free software? Luettavissa: <http://www.gnu.org/philosophy/free-sw.html>. Luettu: 15.7.2017.

Joomla. 2017. Luettavissa: <https://www.joomla.org/about-joomla.html>. Luettu: 19.9.2017

Mening, R. 2015. Wordpres vs Joomla vs Drupal. Luettavissa: <https://website-setup.org/cms-comparison-wordpress-vs-joomla-drupal/>. Luettu: 27.10.2017

Miller, B. 2011. Understanding the Principles of Successful Web Site Design. Luettavissa: <http://media2.fwpublications.com/DESIGN/Digital-Downloads/anatomy-of-a-webpage.pdf?lid=ksfhtmaws083013>. Luettu: 5.11.2017

Robbins Niederst, J. 2006. Web Desing in a Nutshell: A Desktop Quick Reference. 3. painos. O'Reilly Media. Kalifornia.

StatCounter. 2017. Luettavissa: <http://gs.statcounter.com/browser-market-share#monthly-201708-201708-bar>. Luettu: 15.7.2017.

Wodehouse C. 2017. Choosing the Right Software Stack. Luettavissa: <https://www.upwork.com/hiring/development/choosing-the-right-software-stack-for-your-website/>. Luettu 15.9.2017.

Wordpress. 2017. Luettavissa: wordpress.org. Luettu: 14.8.2017.

Wordpress. 2017. Installing WordPress. Luettavissa: https://codex.wordpress.org/Installing_WordPress. Luettu: 15.7.2017.

Opensource. 2017. Luettavissa: opensource.org. Luettu: 5.11.2017.

Liitteet

Liite 1. Osa WordPress-teeman lähdekoodia

header.php

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" <?php language_attributes(); ?>
<head profile="http://gmpg.org/xfn/11">
    <title><?php
        if ( is_single() ) { single_post_title(); }
        elseif ( is_home() || is_front_page() ) { bloginfo('name'); print '
| '; bloginfo('description'); get_page_number(); }
        elseif ( is_page() ) { single_post_title(''); }
        elseif ( is_search() ) { bloginfo('name'); print ' | Search results
for ' . wp_specialchars($s); get_page_number(); }
        elseif ( is_404() ) { bloginfo('name'); print ' | Not Found'; }
        else { bloginfo('name'); wp_title('|'); get_page_number(); }
    ?></title>
    <!--custom-->
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="stylesheet" href="fonts/font-awesome-4.7.0/css/font-
awesome.min.css">
    <!-- -->
    <meta http-equiv="content-type" content="<?php bloginfo('html_type');
?>; charset=<?php bloginfo('charset'); ?>" />

    <link rel="stylesheet" type="text/css" href="<?php bloginfo('sty-
lesheet_url'); ?>" />

    <?php if ( is_singular() ) wp_enqueue_script( 'comment-reply' ); ?>

    <?php wp_head(); ?>

    <link rel="alternate" type="application/rss+xml" href="<?php blo-
ginfo('rss2_url'); ?>" title="<?php printf( __( '%s latest posts', 'hbd-
theme' ), wp_specialchars( get_blogger('name'), 1 ) ); ?>" />
    <link rel="alternate" type="application/rss+xml" href="<?php blo-
ginfo('comments_rss2_url') ?>" title="<?php printf( __( '%s latest com-
ments', 'hbd-theme' ), wp_specialchars( get_blogger('name'), 1 ) ); ?>" />
    <link rel="pingback" href="<?php bloginfo('pingback_url'); ?>" />
</head>
<body>
<div id="wrapper" class="hfeed">
    <div id="header">
        <div id="masthead">

            
```

```

        <div id="access">
            <?php wp_nav_menu( array( 'sort_column' => 'menu_order',
'container_class' => 'menu-header' ) ); ?>
        </div><!-- #access -->

    </div><!-- #masthead -->
</div><!-- #header -->

<div id="main">

```

styles.css

```

@import url('https://fonts.googleapis.com/css?family=Raleway:300,400');

html
{
    margin: 0;
    padding: 0;
    height: 100%;
    width: 100%;
}
body
{
    background: #fff;
    margin: 0;
    height: 100%;
    width: 100%;
    color: white;
}
body, input, textarea
{
    font-family: 'Raleway', sans-serif;
    line-height: 1.5;
    font-size: 18px;
    font-weight: 300;
}
input, textarea{
    color:#000 !important;
}
h1, h2, h3, h4, h5, h6
{
    font-weight: normal;
    clear: both;
}
hr
{
    background-color: #999;
    border: 0;
    height: 1px;
    margin-bottom: 1.5em;
}

```



```

}
pre
{
    font: 11px Monaco, monospace;
    line-height: 1.5;
    margin-bottom: 1.5em;
}
code
{
    font: 11px Monaco, monospace;
}
p
{
    margin: 4px 0 12px 0;
}

/** PAGE LAYOUT **/
/* Styles that affect the entire layout of the page */
div#container
{
    height: 100%;
    width: 90%;
    margin-left: 5%;
    padding-bottom: 15%;
    background: rgba(0,0,0,0.9);
}

div#wrapper{
    height: 100%;
}

/** HEADER : MENU **/
/* To see a menu in your header, you'll need to first add one in WordPress
Admin */
div#header {
    color: black;
    font-size: 50px;
    font-weight: 600;
    font-variant-caps: petite-caps;
    line-height: 1em;
    margin-top: 60px;
    text-align: center;
}

div#access{
    position: fixed;
    top: 0;
    width: 100%;
    background: #000505;
    height: 35px;
}

```

```

        overflow: hidden;
        z-index: 1;
    }
    .menu ul
    {
        list-style: none;
        margin: 0;
        text-align: right;
        padding: 0 1em;
    }

    .menu li
    {
        display: inline-block;
        font-size: 28px !important;
        vertical-align: text-top;
        margin-right: 1em;
        margin-top: -9px;
    }

    .menu-item a
    {
        text-decoration: none;
        text-transform: uppercase;
        color: #484e59;
        text-shadow: 0px 14px 12px #4e4e4e;
    }

    .menu-item a:hover
    {
        color: #2c2d2d;
    }

    .current-menu-item a{
        color: #2c2d2d;
    }

    /** CONTENT : POSTS AND PAGES ***/
    /* Entry and Page Titles */
    h1.entry-title
    {
        display: none;
    }

    /* Media Alignment */
    .alignright
    {
        /* Style for right-aligned images in posts */
        padding: 8px 0 8px 8px;
        float: right;
    }

```

```

.alignleft
{
    /* Style for left-aligned images in posts */
    padding: 8px 8px 8px 0;
    float: left;
}

/* Entry Content */
.entry-content
{
    /* Style that affects actual entry/post content */
    margin: 0 5%;
}

/** CONTENT : COMMENTS */
.comments img
{
    /* Style for the user avatar beside each comment */
    float: left;
    margin: 0 12px 8px 0;
}

/** FOOTER */
div#footer
{
    padding: 1em;
    border-top: 2px solid black;
    text-align: center;
    background: #000505;
    color: #fff;
    position: fixed;
    left: 0px;
    bottom: 0px;
    height: 30px;
    width: 100%;
    font-size: 14px;
}

/** YOUR CSS */
a, a:visited {
    font-variant-caps: all-small-caps;
    text-decoration: none;
    color: white;
}
a:hover{
    text-decoration: underline;
    color: #541093;
}

i.fa{

```

```
        color:white;
        font-size:30px;
    }
    i.fa:hover{
        color:#541093;
    }

    div#site-info {
        display:flex;
    }

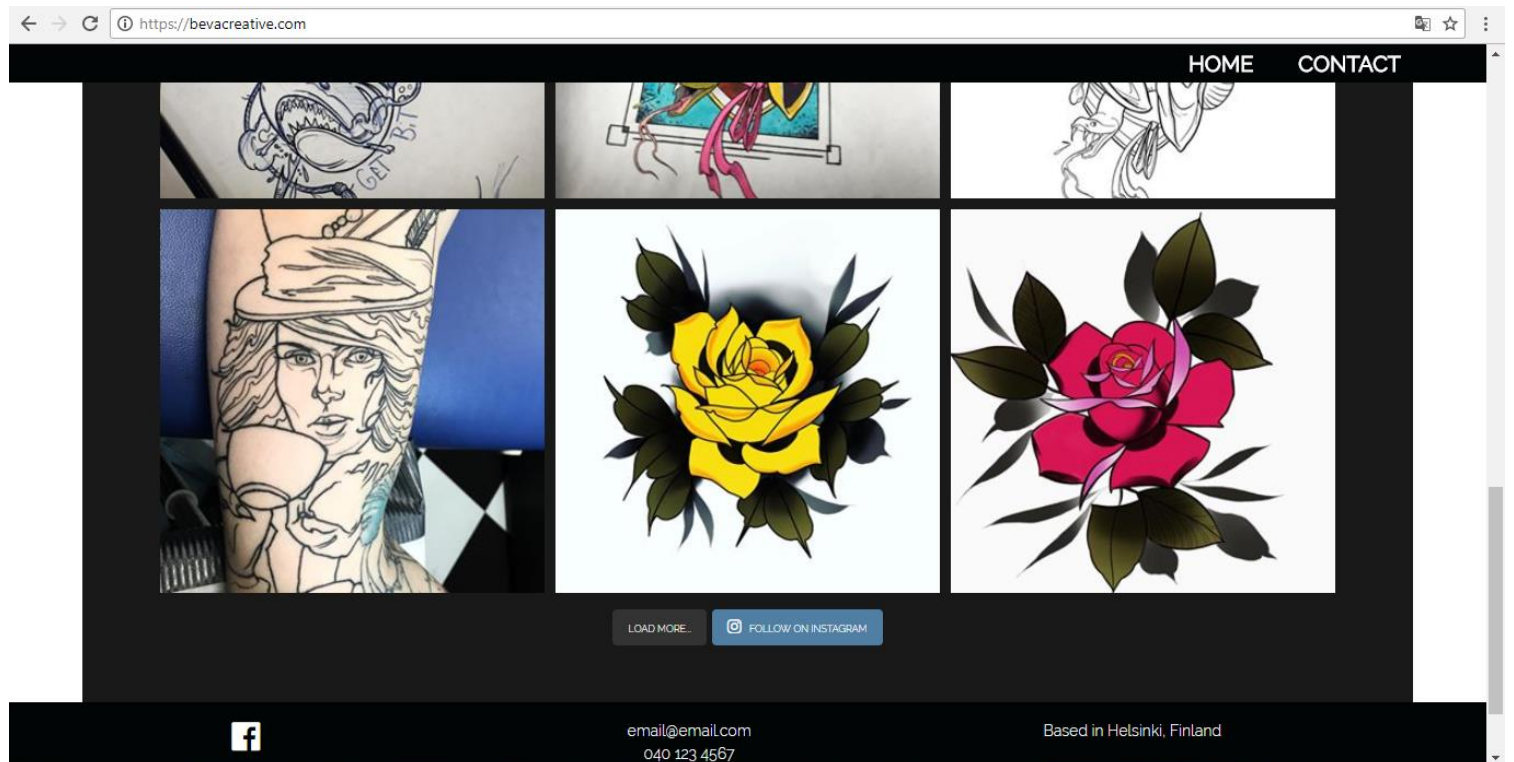
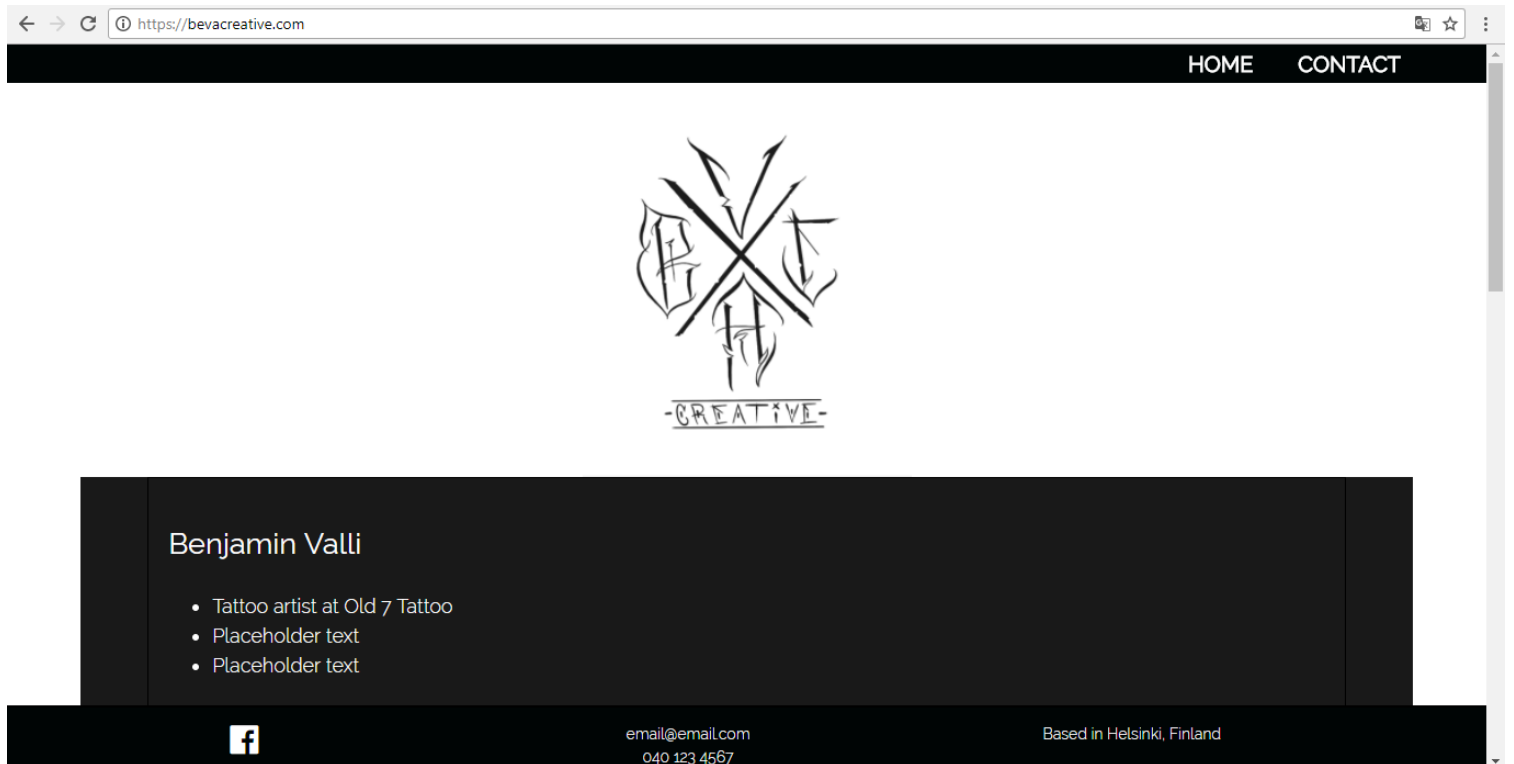
    .footer-item{
        width:30%;
    }

    .footer-item p{
        margin:0;
    }

    .img-round{
        border-radius:106px;
    }

    @media screen and (max-width:959px) {
        div#container {
            width: 98%;
            margin-left: 1%;
        }
    }
}
```

Liite 2. Kuvakaappaukset sivustosta



Turvallinen

https://bevacreative.com/index.php/contact/

HOMECONTACT


Contact me

Your email*

Subject*


Message*

Security Code:



Please enter the security code:

Send



email@email.com

040 123 4567

Based in Helsinki, Finland

27